# Safe Reinforcement Learning for Objects Manipulation in Safety-Critical Coordinated Tasks

**Fawad Khan[1], Wei Feng[2], Zhiyong Wang[1], Tianlun Huang[1], Liu Xiao[1], Asad Ali Shahid[3], Weijun Wang*[1]**

[1]Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China.
[2]Guangdong Provincial Key Lab of Construction Robotics and Intelligent Manufacturing, Shenzhen, China.
[3]Dalle Molle Institute for Artificial Intelligence (IDSIA), Lugano, Switzerland.

fawad, zy.wang, tl.huang1, xiao.liu1@siat.ac.cn, wj.wang@giat.ac.cn, wei.feng@siat.ac.cn, asadali.shahid@idsia.ch

**Abstract -**

**This paper presents a safety-focused, closed-loop grasping approach for object manipulation in dynamic environments, leveraging simulation data to ensure adaptive and safety-aware operations. We extend the OpenAI Safety Gym library by integrating a robotic arm model and propose 'Grasp Mechanics,' a novel method for adaptive gripping. Using a constrained variant of Proximal Policy Optimization (cPPO), we emphasize secure object manipulation in human-robot interaction scenarios. Preliminary experiments show that cPPO, while requiring longer training, achieves policy performance comparable to baseline PPO and significantly outperforms it in safety compliance. Our work highlights the potential of integrating advanced RL techniques with robust safety mechanisms, advancing the capabilities of robotic systems for real-world applications. These findings lay the groundwork for future advancements in safe autonomous robotics, emphasizing the importance of integrating safety considerations from the ground up.**

**Keywords -**

**Safe reinforcement learning, Robotic grasping, Autonomous learning, safety-critical coordination.**

## 1 Introduction

The exploration of safety in Deep Reinforcement Learning (DRL) is critical for advancing autonomous systems. DRL's adaptability and ability to learn through interaction make it a powerful tool for dynamic decision-making, particularly in tasks like robotic manipulation [1]. While RL has historically advanced in simulation environments, simulators often lack the precision needed for seamless real-world transfer [2]. Modeling nuanced behaviors in complex, high-risk systems, such as human-robot interactions, remains challenging [3], requiring sufficient rendering fidelity and simulation realism to capture sensitive dynamics.

Real-world environments present unpredictable challenges, and real-world training carries risks, as errors can lead to unsafe outcomes. Platforms like OpenAI's Safety Gym [4] address these issues by emphasizing safe exploration through predefined constraints, evaluating performance using task success, constraint adherence, and safety violation costs. We adopt these metrics to balance task efficiency and safety compliance.

Building on Safety Gym, we integrate the Panda robotic arm, creating a novel safety-aware environment for object manipulation. Unlike prior work relying on predefined policies [5], our approach combines obstacle avoidance, movement planning, and grasping through learning-based policies. By adapting Safety Gym's constraint mechanisms, we develop customized reward and cost functions that balance task completion with safety.

In this study, we introduce a novel approach for robotic grasping in dynamic environments. Our method employs 3D translation control with a strategically fixed orientation, optimized for grasping common household objects such as a Lipton tea box or a teacup on various surfaces. While full 6D grasping (with dynamic orientation control) presents significant challenges in motion planning and feedback integration, our simplified approach achieves remarkable effectiveness while reducing computational complexity. Unlike typical bin-picking implementations where top-down approaches suffice, our method addresses more complex scenarios through adaptive motion planning and real-time feedback. Previous work has largely focused on grasp pose generation rather than complete motion trajectories, limiting effectiveness in dynamic environments [6, 7]. In contrast, we propose a novel closed-loop grasping and motion planning policy that integrates proprioceptive and tactile feedback, enabling the robot to adapt to real-time object interactions without relying on high-dimensional real-world data (such as images). Our key innovation lies in the hierarchical RL framework [1], which decomposes complex grasping tasks into manageable sub-goals, significantly reducing learning complexity while maintaining robust performance across diverse objects and tasks.

Our research revealed an interesting finding: robots can effectively manipulate objects in dynamic environments without image-based sensory data, suggesting that safety-critical control strategies and adaptive gripping mechanics are more fundamental than visual processing. This insight represents a potential paradigm shift, positioning vision as an enhancement rather than a prereq-

uisite for successful robotic manipulation. This aligns with recent approaches such as the teacher-student training framework [8], where a teacher policy first trained with privileged information then guides a student policy relying on image-based inputs.

A video demonstration of the simulation experiments can be found at: `https://www.youtube.com/watch?v=9oEhdw1F3aI&t=28s`.

## 2 Related Works

Recognizing the challenges of ensuring safe interactions, we draw upon the consensus that safe behavior in robots equates to avoiding hazardous states, a perspective supported by [9] and further elaborated through various safety and learning strategies in the literature. Significant contributions to safety in RL include the development of constraints to prevent unsafe actions, as suggested by [10], and the incorporation of concepts like reachability and stability in algorithm design. Advanced techniques emphasize aligning robot actions with human preferences for enhanced safety [4], explored through ergodicity criteria and the mitigation of adverse "side effects" in agent behaviors. In addressing the intricacies of robotic grasping and task achievement, our study is inspired by and aims, in the future, to rely on Learning from Demonstration (LfD) and RL, with a specific focus on blending these methodologies for optimized task performance. Influential works by [11] demonstrate the potential of merging human demonstrations with RL for nuanced strategy development. This hybrid approach, along with advancements in deep learning, offers new pathways for robotic control and manipulation, particularly in the realm of object grasping where recognizing and handling objects of various geometries remains a complex challenge. Recent advancements in DRL, including PPO and its constrained variant, highlight improved efficiency and safety in robotic tasks. Leveraging these innovations, our research focuses on enabling robots to safely grasp and manipulate objects in dynamic, unstructured environments, mitigating risks in human-robot interaction and advancing autonomy in safety-critical scenarios.

### 2.1 Our Contributions

- **Safety Gym Extension:** Integrated a robotic arm into OpenAI's Safety Gym, creating a novel environment for safe object manipulation.
- **cPPO Advantages:** Through a rigorous comparison with standard PPO, demonstrated cPPO's capability to maintain safety constraints during robotic manipulation, while achieving comparable task performance.
- **Adaptive Grasping Policy:** Developed a closed-loop, goal-directed grasping policy that leverages proprioceptive and tactile feedback for real-time adaptation to varied objects, contrasting with previous approaches that rely on predefined policies or offline planning methods [6].
- **Zero-Shot Generalization:** Achieved effective policy transfer to unseen objects after training on a simple cube.
- **Efficient Learning:** Demonstrated rapid convergence within 200,000 steps while maintaining safety constraints.

Our approach addresses a critical gap in robotic manipulation by maintaining strict collision avoidance with obstacle while preserving task performance, a balance often overlooked in existing systems that prioritize task success metrics while overlooking operational safety considerations.

### 2.2 Finding Technical Solutions

The availability of Python resources and simulators based on various physics engines offers significant advantages but also introduces integration challenges. Our goal was to demonstrate constrained RL algorithms using a robotic arm in human-robot interaction scenarios. However, the Safety Gym library [4] lacks robotic arm agents, posing integration challenges. Initially, we attempted to integrate a robotic arm from a Gym environment into Safety Gym, assuming compatibility due to their shared MuJoCo foundation [12]. However, this revealed dependency conflicts and required older versions of Python and TensorFlow. We also explored integration with CoppeliaSim via PyRep but faced similar compatibility issues. Ultimately, we adopted a PyBullet-based Gym environment [13], which supports robotic arms and provided comprehensive documentation and examples, simplifying the configuration process.

## 3 Problem Formulation

Our research integrates Safety Gym with the PyBullet physics engine, using a Franka Emika Panda robot (7-DoF arm, 2-DoF gripper) for tabletop manipulation tasks around obstacles. The task involves maneuvering objects while avoiding hazards in a scene comprising the robotic arm, an obstacle, and a target object to be placed inside a cabinet.

The task is formulated as a Constrained Markov Decision Process (CMDP) [4], extending standard MDPs with cost functions to balance reward maximization and hazard avoidance [14].

The robot's actions are represented as $[dx, dy, dz, g]$, where $[dx, dy, dz]$ are incremental changes in the end-effector's 3D position, and $g$ is a binary gripper state (open or closed). An inverse kinematics (IK) solver translates these into joint movements for precise operation.

The state is an 8-dimensional vector encoding the end-effector's position, the object's position, and the gripper's status, enhanced by Grasp Mechanics for adaptive and precise manipulation.

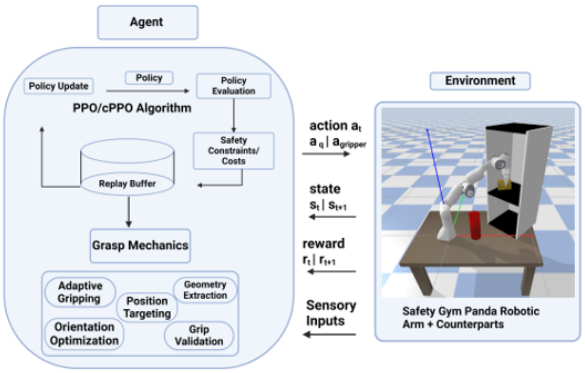


Figure 1. Agent Equipped with Grasp Mechanics

# 4 METHODOLOGY

## 4.1 Safe Reinforcement Learning Through Constraints

RL[4] is an advanced methodology focused on training RL agents to operate within predefined safety constraints, ensuring safe behavior both during training and deployment. An optimal policy in safe RL is defined as:

$$\pi^* = \arg\max_{\pi \in \Pi_C} J_r(\pi) \quad \text{where} \quad \Pi_C = \{\pi : J_{c_i}(\pi) \leq d_i,\ i = 1, \ldots, k\} \tag{1}$$

Where $J_r(\pi)$ represents an objective function driven by rewards and each $J_{c_i}$ signifies a cost-oriented constraint function, with thresholds $d_i$ chosen as hyper-parameters by humans. This Constraint mechanisms define a valid set of permissible policies $\Pi_C$ that meet the constraints, within the framework of CMDPs. The Lagrangian technique employs a composite loss function, integrating both reward-driven and cost-driven elements. In the equation below, cost-related aspect is incorporated within the set of allowable policies $\Pi_C$. This optimization task is formulated as follows:

$$\max_{\theta} \min_{\lambda \geq 0} L(\phi, \theta) = f(\theta) - \lambda g(\theta) \tag{2}$$

Within the loss function, the two components pertain to the reward and cost, incorporating the policy model parameters $\theta$ and the Lagrangian hyper-parameter $\lambda$.

### 4.1.1 Pre-Learning Phase

The pre-learning phase encodes task and safety requirements while establishing an instrumented training environment. Key components include:

**Intended Functionality:** Unlike traditional software, the behavior of machine learning-based skills cannot be fully defined or verified beforehand [15]. Following ISO/PAS 21448:2019, we use "intended functionality" to define requirements, establishing boundaries for task performance and safety.

**Expert Knowledge:** Expert input encodes operational context aligned with real-world applications, embedding task requirements via reward functions and safety constraints [16], defining action spaces leveraging existing robot skills [17], establishing state spaces with necessary perception capabilities, and incorporating demonstrations to accelerate learning [18].

We partially automate the encoding process using hierarchical RL, where sub-goals guide the agent toward specific sub-tasks. The environment uses standard robot description models (URDF), and the RL agent is trained to identify hazardous states while maintaining performance.

### 4.1.2 State Space

Our framework integrates simulation data with proprioceptive state information, including robot states (forces on the end-effector), fingertip states, object states, force sensor data, object geometry extraction, and distance measurements. These elements provide tactile feedback for adaptive gripping, ensuring a continuous state space essential for effective manipulation task learning. In continuous state spaces, deep neural networks (DNNs) excel at learning low-dimensional representations that map states to optimal control actions.

### 4.1.3 Action Space

In Human-Robot Collaboration (HRC) scenarios, the action space is defined by complex interactions in a six-dimensional space (the robot's end-effector pose). This continuous action space poses significant challenges for RL agents, often requiring millions of time steps for training, even with robust exploration and exploitation algorithms [19]. To address this, recent approaches constrain the action space by leveraging Inverse Kinematics (IK) solvers instead of directly learning joint angles [20].

This method reduces the action space dimensionality from seven (corresponding to the robot arm's seven degrees of freedom) to three, focusing only on positional changes in Cartesian space ($\Delta x, \Delta y, \Delta z$) with fixed orientations. This reduction accelerates learning, enhances task efficiency and safety, and promotes robust output behaviors by minimizing unintended actions. Furthermore, it facilitates the transferability of learned behaviors across different robots, improving generalization and adaptability in diverse HRC applications.

### 4.1.4 Reward Function

The reward function guides the robotic arm in object manipulation tasks, ensuring precise movement and obstacle avoidance. Negative reward functions act as soft penalties, which may not guarantee safety. Thus, Constrained RL maximizes expected returns while adhering to safety constraints using cost functions [4]. Designing effective reward functions constitutes approximately

**60%** of development effort, directly influencing learning efficiency and task performance.

### 4.2 Hierarchical RL Framework

In recent years, Hierarchical RL (HRL) has emerged as a powerful framework for solving complex decision-making tasks by decomposing them into a hierarchy of simpler subtasks [1]. This approach is particularly beneficial for robotic manipulation tasks, where a robot must perform a sequence of actions with long-term dependencies and complex goals. In this work, we apply HRL to a robotic arm tasked with manipulating objects in an environment. The task is divided into phases approaching, grasping, and lifting each governed by a low-level policy, while the high-level policy coordinates the transitions between these phases, with each phase having its corresponding reward function to guide learning. This hierarchical structure enhances both learning efficiency and execution performance by allowing the robot (agent) to focus on simpler sub-problems sequentially.

## 5 IMPLEMENTATION SETTINGS

We employ the PPO algorithm with the PPO-Clip variant [21] to enhance policy gradient stability and efficiency.

To address safety concerns within the RL framework, our work integrates the PPO algorithm with the Lagrangian method for constrained optimization. This integration allows us to define a feasible set of policies ($\pi c$) that satisfy predefined safety constraints $J_{ci}(\pi) \leq d_i$, CMDPs incorporate a set of cost functions $c_1, ....c_k$, which are distinct from the reward function, as outlined in the framework of CMDPs [4] (see subsection 4.1 in Methodology). In our scenario, if the agent makes contact with an obstacle at time step $t$, it incurs a cost of $c(s_t) = 1$; otherwise, the cost is 0. This is represented by $d_i = 1$ if the robotic arm violates the safety constraint, and $d_i = 0$ otherwise. This is essential to constrain hazardous behavior within the Safety Gym environment. Our integration of PPO and its Lagrangian extension advances both policy optimization and safety-aware human-robot interactions, ensuring robotic agents operate within safety margins in simulated environments, a foundation for constrained RL solutions applicable to real-world scenarios.

### 5.1 Multi-step Dense Reward System

This section describes a multi-step dense reward system for training agents in object manipulation tasks, providing continuous feedback to guide the arm through approaching, grasping, and lifting phases.

The core of the reward system is a distance-based reward, incentivizing the agent to minimize the Euclidean distance between the end-effector and the target object: $\text{dis} = \sqrt{(x_r - x_d)^2 + (y_r - y_d)^2 + (z_r - z_d)^2}$, with $r_{\text{distance}} = -\min(\text{dis} \cdot 0.1, 5)$. This design, grounded in reward shaping [22], ensures incremental improvement in positioning. Additionally, a contact stability reward encourages stable grasping: $r_{\text{contact}} = 2$ for first contact with both fingers, 3 for maintaining stable contact, $-2$ for persistent single-finger slip, and $-1$ for losing established contact.

Once the object is lifted above a threshold, a significant positive reward reinforces the desired behavior: $r_{\text{grasp\_lift}} = 5$. Safety is ensured through collision detection: $c_{\text{collision}} = \mathbb{1}(\text{contact between robot and obstacle})$, with info = $\{\text{cost} : c_{\text{collision}}\}$, aligning with CMDPs [23] to evaluate both task performance and safety.

### 5.2 Model Structure and Training

Our method employs two RL algorithms: PPO [21] and its Lagrangian variant. The agent explores the environment through on-policy learning, collecting data via controlled policy rollouts rather than random exploration. While we use actor-critic algorithms, our approach is generalizable to any deep Q-learning algorithm, whether online or offline.

**PPO**: An on-policy actor-critic algorithm that optimizes cumulative rewards using a clipped surrogate objective, known for its simplicity and stability in continuous control tasks.

**PPO Lagrangian Variant**: Extends PPO by incorporating a Lagrangian multiplier to enforce safety constraints during training, making it ideal for safety-critical tasks. Both algorithms utilize low-dimensional state information and tactile feedback, enhanced by Grasp Mechanics for adaptive manipulation.

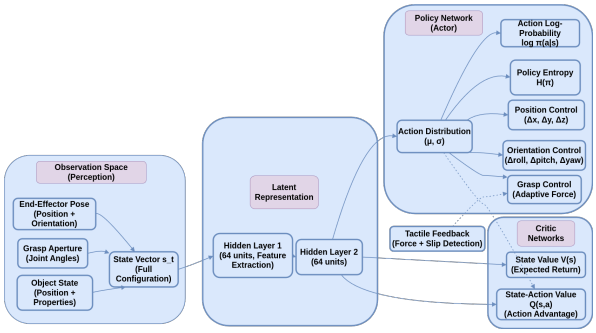


Figure 2. Actor-Critic Architecture for Robotic Manipulation with Tactile Feedback.

The neural network architecture consists of an 8-dimensional input, two hidden layers with 64 neurons each, and a 4-dimensional output for actions. Training dynamics involve 200 epochs, each comprising 1000 steps, with episodes capped at 500 steps. The policy network uses an MLP Gaussian policy for continuous actions, calculating actions, log probabilities, and optimization metrics. Value networks estimate state and state-conditioned values to ensure accurate policy evaluation. Key hyperparameters include: `vf_lr` ($1 \times 10^{-3}$), `penalty_lr` ($5 \times 10^{-2}$), `gamma` (0.99),

`cost_gamma` (0.99), `lam` (0.97), `cost_lam` (0.97), `ent_reg` (0.01), `cost_lim` (25), `penalty_init` (1.0), `target_kl` (0.01), and `vf_iters` (80). Optimization techniques include 80 gradient descent iterations per update cycle, a clipping ratio to prevent drastic policy changes, and backtracking steps for efficient optimization.

**Algorithm 1:** Constrained PPO

```
Data: Environment (env)
Result: Trained Policy
Initialization: Initialize logger, seeds,
  environment, buffers, parameters,and
  optimizers.
Policy Learning: Compute likelihood ratio,
  surrogate advantage, and cost.
Main Loop: Initialize environment variables.
Epoch Loop: for each epoch do
    Update penalty;
    for each step do
        Render env, get action, step in env;
        Store experience, update stats;
        if terminal state or end of epoch then
            Finish path, reset env;
        end
    end
    Save model, update policy and value
      functions;
    Log performance and stats;
end
Main Function: Parse args, initialize logger and
  agent. Execute
  run_polopt_agent(parameters).
```

### 5.3 Grasping Policy via Grasp Mechanics

Our research focuses on learning a closed-loop policy for robotic grasping, enabling autonomous object manipulation. The policy maps the current state $s_t$ at time $t$ to an action $a_t$, primarily involving 3D translation of the robot's end-effector while maintaining an optimized fixed orientation. To simplify grasping, we leverage 'Grasp Mechanics' using PyBullet utilities, integrating tactile feedback from PyBullet's `getVisualShapeData` function and force sensors. This provides detailed visual attributes (e.g., shape type and dimensions) essential for precise interactions within our RL framework.

'Grasp Mechanics' combines five key processes for nuanced and safe object interaction: (1) **Adaptive Gripping**, which dynamically adjusts grip based on proximity and contact force; (2) **Geometry Extraction**, retrieving spatial characteristics for robust recognition; (3) **Orientation Optimization**, aligning the gripper for maximum stability; (4) **Position Targeting**, ensuring precise gripper placement; and (5) **Grip Validation**, assessing grip security before lifting. These functionalities, integrated with Safety Gym capabilities, form the backbone of our approach, enabling high-precision, adaptable robotic manipulation.

### 5.4 Task Success and Evaluation Metrics

Task success is evaluated using three metrics from the Safety Gym protocol [4]: (1) **Task Performance**, measured as the percentage of successfully completed tasks (approaching, grasping, and lifting) out of total attempts (Figure 3), Success Rate $= \frac{\text{Successful Tasks}}{\text{Total Tasks}}$; (2) **Constraint Satisfaction**, tracking safety violations (e.g., collisions), with fewer violations indicating better safety; and (3) **Average Regret**, reflecting cumulative safety costs during training, Average Regret $= \frac{1}{N}\sum_{i=1}^{N} C_i$, where $C_i$ is the safety cost at step $i$, and $N$ is the total number of steps. These metrics balance task efficiency and safety compliance.

### 5.5 Zero-shot Policy Transfer Across Objects

We trained our robotic manipulation system through 500,000 iterations using a cube as the reference object. The system's MLP actor-critic architecture processed state vectors containing object pose, gripper state, and spatial relationships. To evaluate generalization capabilities, we tested the trained policy on unseen objects: a Lipton tea box, teacup, soap bar, and remote control. The system successfully grasped all test objects, with highest success rates for the Lipton tea box 95% and soap bar 90%. While performance remained robust on objects with regular geometries, success rates decreased slightly for the teacup 87% and remote control 85% due to their complex or asymmetrical shapes challenging gripper alignment. The system demonstrated effective zero-shot transfer of manipulation policies without object-specific training.

#### 5.5.1 Performance Metrics

The policy achieved a 99% success rate on training objects and demonstrated strong generalization with 95% and 90% success rates on unseen objects like the Lipton tea box and soap bar. Performance decreased slightly with geometrically complex items such as teacups and remote controls. Objects with regular, symmetrical shapes yielded higher success rates and faster completion times, as shown in Figure 3. Conversely, items requiring precise gripper alignment due to non-uniform surfaces or irregular edges presented greater challenges, often leading the agent to make multiple attempts to achieve a firm grasp. Performance correlated positively with symmetry and graspable edges, while surface complexity inversely affected success rates. These results highlight the system's effectiveness with standard objects while identifying improvement opportunities for handling complex geometries under safety constraints and time limitations. Further policy refinement is needed to ensure robust manipulation across a wider range of irregular or non-uniform objects.

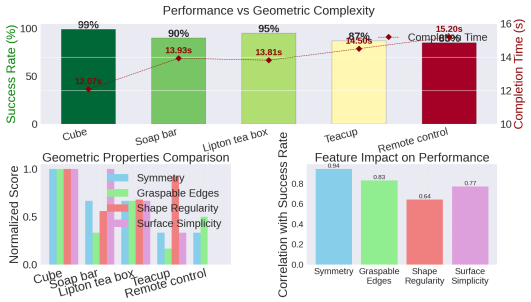


Figure 3. Performance evaluation of zero-shot policy transfer across objects with varying geometries, demonstrating the system's ability to generalize and highlighting the impact of features like symmetry, graspable edges, shape regularity, and surface simplicity on success and completion time. Note: Results averaged over 40 attempts per object.

### 5.5.2 Comparative Analysis

We compared our closed-loop grasping approach to object manipulation, which fundamentally differs from traditional fixed-policy methods [7]. Unlike these fixed-sequence approaches [6],[7], our system utilizes a purely learning-based framework that dynamically adapts within a sequential decision-making architecture. The key innovation lies in its ability to monitor and adjust behavior based on the current state. For example, in failure scenarios, such as when a grasp attempt fails and the object is not secured, the system reassesses and retries, rather than continuing with the predefined sequence. As shown in Figure 3 and Table 1, our approach achieves higher success rates and adapts more effectively to varying object shapes and positions.

**Comparison with Existing Approaches**

We empirically compared our approach with baselines across several key features as shown in Table 1.

Table 1. Comparison of Object Grasping and Manipulation Approaches

| Feature | Our Approach | Baselines [7, 6] |
|---|---|---|
| Learning Paradigm | Goal-Directed Reinforcement Learning | Imitation Learning (predefined behavior + learning from data) |
| Decision Making | Sequential, state-dependent with learned policy | Fixed action sequences with predefined policy |
| Failure Handling | Dynamic reassessment with real-time adaptation | Continue with predefined sequence, static behavior |
| State Awareness | Continuous monitoring of gripper state | Limited state feedback |
| Task Execution | Safety-aware operation while maintaining performance | Performance optimization without safety considerations |

## 6 EXPERIMENT DESIGN AND REALIZATION

Our analysis of robotic arm control in 3D Cartesian space reveals that increasing the state vector's dimensionality does not significantly improve learning efficiency for adaptive grasping. However, integrating 'Grasp Mechanics' as a feedback mechanism enhances both learning efficiency and overall effectiveness. A comparative study of PPO and its constrained variant, cPPO, shows that cPPO incorporating Lagrangian safety methods achieves lower average costs, indicating safer training despite slightly longer training periods. Figure 4 illustrates this trade-off, comparing reward values and costs across training cycles. The graph differentiates between average cumulative rewards and episode-wise costs for PPO and cPPO, demonstrating the impact of constrained optimization on safety and effectiveness in robotic control.

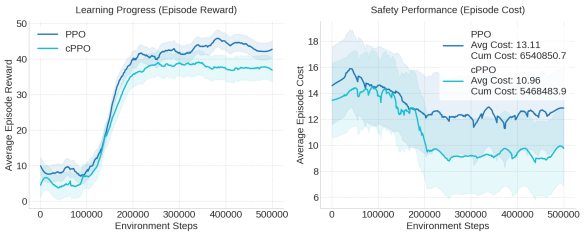


Figure 4. PPO vs. cPPO Rewards and Costs Comparison in Object Manipulation.

The visual comparison highlights cPPO's gradual learning curve and consistently lower cost metrics, reflecting its prioritization of safety. This systematic minimization of costs ensures safer operational dynamics for the robotic arm.

The Table 2. compares the average costs per run for the standard PPO algorithm and its constrained variant (cPPO). While PPO incurs an average cost of 13.11, cPPO achieves a lower cost of 10.96, reflecting its enhanced safety performance. This evaluation is based on a task involving the manipulation of various geometries and securing them at target locations under strict safety constraints, using a 3D action representation in Cartesian space.

Table 2. Average comparison costs per run for PPO and cPPO.

| Algorithms | Average Cost Incurred |
|---|---|
| PPO | 13.11 |
| cPPO | 10.96 |

### 6.1 Discussion and Conclusion

This paper addresses the challenge of ensuring safe robotic manipulation in dynamic environments, where

robots must interact with diverse objects while adhering to safety constraints. Our approach leverages cPPO to develop policies that balance task performance with safety. We introduce Grasp Mechanics, a novel closed-loop grasping policy that combines compact state information acquisition with precise positional control. By relying primarily on proprioceptive, low-dimensional kinematic data, our system achieves effective manipulation without requiring detailed object shape information, a key advantage in real-world environments where high-dimensional data, such as images, may be unavailable.

Our experimental results validate two key contributions. First, the cPPO implementation demonstrates consistent improvement in safety performance, reducing the average episode cost from approximately 14 to 10.96 while maintaining task effectiveness. This reduction in safety cost, combined with the stable learning progress shown in the reward curve, indicates that our approach successfully balances safe navigation with task completion. Second, the system demonstrates an 89% success rate in grasping previously unseen objects using a purely learning-based policy, highlighting that safe manipulation can be achieved without compromising task effectiveness. In contrast, previous work relied on predefined policies [5], which may have limited flexibility and adaptability.

The learning curves from our experiments reveal both strengths and limitations of the current approach. While cPPO demonstrates consistent convergence to safe policies within 200k training steps, we observed that the initial exploration phase requires careful tuning of safety boundaries to prevent constraint violations. This highlights a trade-off between learning efficiency and safety guarantees that warrants further investigation.

#### 6.1.1 Limitations and Future Work

Our approach has two key limitations. First, the use of a fixed orientation strategy, rather than full 6D control, restricts the ability to execute orientation-sensitive grasps, particularly for objects with asymmetric affordances (e.g., a shampoo bottle with the pump facing upward), which cannot be effectively grasped from a top-down view. While this design choice reduces learning complexity, it still achieves practical effectiveness for a wide range of household objects. Second, reliance on proprioceptive feedback alone, though computationally efficient, may hinder performance in scenarios requiring visual feedback for precise manipulation.

Despite these limitations, our results demonstrate that proprioceptive feedback enables robust grasping while maintaining safety constraints. The fixed orientation approach, though not fully exploiting all degrees of freedom, proves effective for common household objects and provides a foundation for future extensions. Two promising directions for future work include: (1) expanding to full 6D control using hierarchical policies that first master 3D positional control before incorporating dynamic orientation adjustments, and (2) implementing a teacher-student framework [8], where our safe manipulation policies serve as teachers for training vision-based students. This approach could reduce the computational cost of learning from visual inputs while maintaining safety, with significant implications for real-world applications. Our research contributes to the field of autonomous robotics by demonstrating that effective object manipulation can be achieved while strictly adhering to safety requirements. The combination of constrained RL and proprioceptive feedback provides a particularly promising foundation for future sim-to-real transfer. Unlike vision-dependent approaches that often struggle with the reality gap due to visual discrepancies, our proprioceptive-focused policy relies on physical interactions and low-dimensional state representations that theoretically transfer more reliably across domains. This design choice deliberately addresses a critical challenge in robotics: policies that perform well in simulation often degrade significantly when deployed on physical hardware. By prioritizing contact dynamics, force-based feedback, and physics-grounded policy constraints, our approach is inherently more resilient to the domain transfer challenges that typically hinder real-world deployment. Future work will focus on systematically bridging the simulation-reality gap through domain randomization and progressive transfer techniques, with particular emphasis on preserving safety guarantees while improving sample efficiency across the full spectrum of manipulation tasks from controlled laboratory environments to dynamic human-centered settings.

## Acknowledgment

This work was supported in part by the National Key R&D Program of China (No. 2023YFB4705002), in part by the National Natural Science Foundation of China (U20A20283), in part by the Guangdong Provincial Key Laboratory of Construction Robotics and Intelligent Construction (2022KSYS013), in part by the CAS Science and Technology Service Network Plan (STS) - Dongguan Special Project (Grant No. 20211600200062), and in part by the Science and Technology Cooperation Project of Chinese Academy of Sciences in Hubei Province Construction 2023.

## References

[1] Benjamin Beyret, Ali Shafti, and A Aldo Faisal. Dot-to-dot: Explainable hierarchical reinforcement learning for robotic manipulation. In *IROS*, pages 5014–5019. IEEE, 2019.

[2] Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Lee, Jie Tan, and Sergey Levine. Learning agile robotic locomotion skills by imitating animals. *arXiv preprint arXiv:2004.00784*, 2020.

[3] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew,

Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

[4] Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. *arXiv*, 2019.

[5] Lirui Wang, Yu Xiang, Wei Yang, Arsalan Mousavian, and Dieter Fox. Goal-auxiliary actor-critic for 6d robotic grasping with point clouds. In *CoRL*, pages 70–80, 2022.

[6] Lirui Wang, Yu Xiang, and Dieter Fox. Manipulation trajectory optimization with online grasp synthesis and selection. *arXiv preprint arXiv:1911.10280*, 2019.

[7] Joao Carvalho, An T Le, Philipp Jahr, Qiao Sun, Julen Urain, Dorothea Koert, and Jan Peters. Grasp diffusion network: Learning grasp generators from partial point clouds with diffusion models in so (3) xr3. *arXiv preprint arXiv:2412.08398*, 2024.

[8] Tao Chen, Jie Xu, and Pulkit Agrawal. A system for general in-hand object re-orientation. In *CoRL, pages=297–307, year=2022, organization=PMLR*.

[9] Alexander Hans, Daniel Schneegaß, Anton Maximilian Schäfer, and Steffen Udluft. Safe exploration for reinforcement learning. In *ESANN*, pages 143–148, 2008.

[10] Moshe Haviv. On constrained markov decision processes. *Operations research letters*, 19(1):25–28, 1996.

[11] Yuke Zhu, Ziyu Wang, Josh Merel, Andrei Rusu, Tom Erez, Serkan Cabi, Saran Tunyasuvunakool, János Kramár, Raia Hadsell, Nando de Freitas, et al. Reinforcement and imitation learning for diverse visuomotor skills. *arXiv preprint arXiv:1802.09564*, 2018.

[12] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[13] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning, 2016.

[14] Marvin Rausand. *Reliability of safety-critical systems: theory and applications*. John Wiley & Sons, 2014.

[15] Rick Salay, Rodrigo Queiroz, and Krzysztof Czarnecki. An analysis of iso 26262: Using machine learning safely in automotive software. *arXiv preprint arXiv:1709.02435*, 2017.

[16] Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, Todd Hester, Cosmin Paduraru, and Yuval Tassa. Safe exploration in continuous action spaces. *arXiv preprint arXiv:1801.08757*, 2018.

[17] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

[18] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel Van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 37(4):1–14, 2018.

[19] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[20] Mohamed El-Shamouty, Xinyang Wu, Shanqi Yang, Marcel Albus, and Marco F Huber. Towards safe human-robot collaboration using deep reinforcement learning. In *ICRA*, pages 4899–4905. IEEE, 2020.

[21] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1–12. PMLR, 2017.

[22] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pages 278–287, 1999.

[23] Eitan Altman. *Constrained Markov decision processes*. Routledge, 2021.